

# MATLAB, v. 6.5.0180913, Release 13

## 1. Úvod

Jedná se o programový systém, jehož název znamená **MATRIX LABORATORY**. Používá se od roku 1984 v mnoha oborech k simulacím, měření, grafice. Používá se celosvětově na univerzitách k výuce matematiky a inženýrských oborů, ve vědeckých institucích, ale také v průmyslu při výzkumu, vývoji a analýze dat. Je to tzv. **interpret**. Umožňuje exportovat obrázky do textových editorů (v adresáři NOTEBOOK je podpora pro MS Word). Pomocí **MATLAB C/C++ Compiler, Math Library a Graphics Library** je možné automaticky překládat programy v jazyce MATLAB do jazyka C nebo C++. Dovoluje vytvářet samostatné aplikace pro všechny podporované platformy. **MATLAB Web Server** dovoluje začleňovat funkce MATLABU do HTML dokumentů. Kromě základního modulu má řadu specializovaných částí, které se nazývají **toolboxy**. Jsou to knihovny funkcí zaměřené na konkrétní technické a vědní obory. Vyžadují specifické znalosti z daného oboru. V předmětu Elektrické obvody 3 budeme používat kromě základní části ještě **Signal Processing Toolbox** a **Simulink** (pracuje s funkcemi a příkazy jako s grafickými bloky, používá se např. pro časové řešení soustavy nelineárních diferenciálních rovnic s grafickým zadáváním řešené soustavy, k simulaci dynamického chování).

MATLAB je možné používat v operačních systémech MS Windows i Linux, SUN, HP, Power Macintosh. Minimální HW konfigurace - paměť 256MB (doporučeno 512 MB) RAM, 16-bit grafická karta, 500 MB diskového prostoru pro hlavní modul MATLAB.

Následující řádky nejsou úplným manuálem ani dostačující učebnicí. Poskytnou pouze stručný úvod do práce s MATLABem.

Matlab slouží k

- vědeckotechnickým výpočtům,
- modelování,
- simulaci, analýze a vizualizaci dat,
- měření a zpracování signálů,
- vývoji algoritmů,
- návrhu řídicích a komunikačních systémů.

MATLAB je vhodné používat pro robustní výpočty nebo pro jednorázové zpracování datových souborů, když je možné převést řešení na vektorové a maticové operace. K řešení se používají **iterační algoritmy** (založené na opakovaném používání vzorců, viz např. Numerické metody ve skriptech Čmela,R., Havlíček,V.,Zemánek,I.: Základy teorie obvodů 1. Cvičení).

MATLAB je nevhodné používat na pomalých počítačích s malou operační pamětí a tam, kde je potřebná především rychlost výpočtů.

Tato verze je objektově orientována. Matematické operace jsou založeny na **maticovém počtu v oboru reálných a komplexních čísel**.

Obsahuje

- běžné matematické funkce
- složitější matematické funkce (Besselovy funkce, transformace souřadnic,

Fourierova transformace, polynomy, numerické integrace a derivace, minimum funkce více proměnných, diferenciální rovnice 1. řádu atd).

## 2. Pracovní plocha

**Command Window** - pro zapisování příkazů a povelů, zobrazování systémových hlášení.

**Current Directory** – zobrazuje aktuální složku. Implicitně je nastavena cesta *MATLAB6p1/Work*. Cestu je možné nastavit také do složky, kam chcete uložit práci.

Z položky **View** v hlavním menu lze spustit následující tři položky:

**Launch Pad** - spouští nabídky, demonstrační programy *dem*os, podporuje tvorby grafických prostředí. Je to interaktivní *help*.

**Command History** – zobrazuje všechny již použité příkazy, které lze opět aktivovat (poklepnutím nebo přetažením myši).

**Workspace** – udává přehled všech použitých proměnných (na začátku práce je prázdné).

Pomocí tlačítek na liště je možné vytvořit nebo otevřít již existující tzv. **m-soubor**, otevřít Simulink a pod.

**m-soubor** je textový soubor s povinnou příponou \*.m. Jedná se o zdrojový kód. Je možné ho vytvořit pomocí nabídky

**File, New, M-file** z nabídky v hlavním menu (také přímo pomocí kláves **Ctrl+N**) nebo v libovolném textovém editoru (jako ASCII, např. NOTEPAD, WORKPAD, apod.).

## 3. Základní operace s čísly z příkazové řádky

Píšeme vždy desetinnou tečku!

Implicitní formát má 4 desetinná místa (výpočty jsou prováděny s *double precision*).

Implicitní proměnná má název *ans* (je možné změnit).

Každý příkaz nebo výraz je třeba potvrdit pomocí **ENTER**.

Ukončení práce je pomocí **quit** nebo **X** v pravém horním rohu okna.

Systém rozlišuje malá a velká písmena!

Příkazy oddělené středníkem je možné zapisovat na jeden řádek.

### a) reálná čísla

#### **Základní elementární funkce:**

<b>sin</b>	sinus
<b>sinh</b>	hyperbolický sinus
<b>asin</b>	inverzní funkce k sinu
<b>asinh</b>	inverzní hyperbolický sinus
<b>cos</b>	kosinus
<b>cosh</b>	hyperbolický kosinus
<b>acos</b>	inverzní funkce ke kosinu
<b>acosh</b>	inverzní hyperbolický kosinus
<b>tan</b>	tangens

---

<b>fix</b>	zaokrouhlení směrem dolů
<b>rem</b>	zbytek po dělení
<b>sign</b>	znaménková funkce
<b>round</b>	zaokrouhlení k nejbližšímu číslu

<b>exp</b>	exponenciální funkce
<b>log</b>	přirozený logaritmus
<b>log10</b>	dekadický logaritmus
<b>sqrt</b>	druhá odmocnina
<b>pow2</b>	mocnina se základem 2 (např. $2^6$ )

**Základní operátory:** +, -, \*, /, ^ (plus, minus, násobeno, děleno, umocněno)

Pomocí šipky **↑** lze vyvolat předchozí příkazy (opakováním použitím), další možností je pomocí *Command History*.

**help** stručné vysvětlení syntaxe případně chybové hlášení.

Např. **help sin**

Pro vypsání názvů elementárních funkcí slouží příkaz **help elfun**

Pro přehled operátorů a speciálních znaků slouží příkaz **help ops**

Pro vypsání možných formátů slouží příkaz **help format**.

#### b) komplexní čísla

Povolené zápisy:  $4+i*3$

$4+3i$

$4+3*i$

$4+3j$

**Základní operace s komplexními čísly (výběr): **help elfun****

<b>abs</b>	absolutní hodnota
<b>angle</b>	výpočet fáze
<b>imag</b>	imaginární část komplexního čísla
<b>real</b>	reálná část komplexního čísla

## 4. Použití proměnných

Vyčistění plochy okna při zachování hodnot všech proměnných: **clc**

Příklad zápisu: **napeti2 = Umax\*sin(omega\*t + fi)**

Jména proměnných mohou obsahovat písmena i číslice, první znak musí být písmeno!

Pro potlačení výpisu obsahu proměnných nebo výrazů v m-souboru slouží **středník ;**

Všechny operace používané pro čísla je možné použít i pro proměnné.

**DOPORUČENÍ:** Přiřadte proměnným číselné hodnoty a výraz zapisujte pomocí proměnných. Tento postup je přehlednější a jednodušší.

*Příklad:*  $2*pi*1000 / (exp(0.2)*abs(2+3i))$

$a=2*pi*1000;$

$b=exp(0.2);$

```
c=abs(2+3i);
XY=a/(b*c)
```

**Vymazání nepotřebných proměnných** (šetří paměť) pomocí **clear**.

**Výpis použitých proměnných** pomocí **who**, **whos** - v rozšířené formě

**Uložení** pomocí **save**

Po ukončení MATLABu jsou neuložená data ztracena. Uložit je můžeme několika způsoby.

a) Uložení ve specifickém formátu **\*.mat**.

<b>save filename1</b>	vše, co obsahuje <i>Workspace</i> se uloží do souboru, který má jméno 'filename1'
<b>save filename2 A</b>	do souboru 'filename2' se uloží pouze <i>A</i>
<b>save filename3 A B C</b>	do souboru 'filename3' se uloží <i>A, B, C</i>
<b>save filename2 D -append</b>	do souboru 'filename2' se uloží ke stávající proměnné <i>A</i> další nová proměnná <i>D</i>

Přípona *mat* je implicitní, není nutné ji psát. Takto uložená data je možné načítat pouze v prostředí MATLAB. Po načtení souboru jsou data přístupná opět v proměnných, které jsme ukládali.

b) Uložení ve formátu **ASCII**

<b>save filename A -ascii</b>	uložení proměnné <i>A</i> do souboru <i>filename</i> v prac. adresáři
<b>save filename A -ascii -tabs</b>	uložení proměnné <i>A</i> do souboru <i>filename</i> ve tvaru tabulky v prac. adresáři
<b>save c:\prac\ filename A -ascii</b>	uložení proměnné <i>A</i> do souboru <i>filename</i> v jiném než pracovním adresáři

Oproti způsobu a) jsou data po načtení přístupná v proměnné, která má jméno shodné se jménem souboru.

Takto uložená data je možné zpracovávat ve všech editorech a programech. Tento způsob je vhodný pro vstupní a výstupní data, zejména u rozsáhlejších a složitějších aplikací. Pro potřeby předmětu EO3 je vhodnější způsob a).

Více informací najdete pomocí **help save**.

**Načtení proměnných** příkazem **load**.

Načtení proměnných nebo souboru je nutné při novém spuštění MATLABu nebo při vymazání proměnných z pracovní plochy. Více viz příkaz **help load**.

**Relační operátory** - viz **help ops**.

<b>==</b>	<i>Equal</i>	je rovno
<b>~=</b>	<i>Not equal</i>	není rovno
<b>&lt;</b>	<i>Less than</i>	menší než
<b>&gt;</b>	<i>Greater than</i>	větší než
<b>&lt;=</b>	<i>Less than or equal</i>	menší nebo rovno
<b>&gt;=</b>	<i>Greater than or equal</i>	větší nebo rovno

## Práce s řetězci

Text je třeba umístit **mezi znaky apostrofu 'text'**. Operace s řetězci jsou obdobné jako u číselných proměnných, existují však i speciální operace.

## Nápověda

**Důležité!!!** Při instalaci MATLABu, v. 6.5.0180913, Release 13 na počítačích s operačním systémem WIN98 nelze využít klávesu **Help** z nabídky. Dostupné je používání dříve uvedeného způsobu nápovědy pomocí **help funkce**. Znamená to, že je třeba znát jména hledaných funkcí (viz dále). Je však možný náhradní způsob. Na počítačích v laboratoři 802 lze úplné texty nápovědy získat na adresách

C:\MATLAB6p5p1\help\techdoc\learn\_matlab\learnina.html  
C:\MATLAB6p5p1\help\techdoc\learn\_matlab\learninng.html

## 5. Práce s maticemi a řešení soustavy rovnic

Systém MATLAB je orientován maticově.

### Vytvoření matic a vektorů

**matice** se zapisují po řádcích, řádky jsou oddělené středníkem, prvky v řádku jsou oddělené mezerou (nejčastěji) nebo čárkou (méně často)

**vektory** se zapisují jako matice s 1 řádkem nebo 1 sloupcem

Počet řádků ani sloupců není omezen.

Tento způsob je vhodný pro matice resp. vektory malých dimenzí (malý počet řádek a sloupců).

Příklad:

```
matice=[1 2 3; 4 5 6; 7 8 9]
vector=[11 12 13 14]
A=[2*log(10) exp(-0.1); sin(pi); 2+3i 0]
```

Pro definování větších matic je vhodný **zápis s dvojtečkami**.

Příklad:

**osa=1: pi/10 : 2\*pi** tento zápis znamená definici vektoru představujícího např. osu začínající v bodě 1 a končící v bodě o hodnotě  $2\pi$  s krokem  $\pi/10$

### Zvláštní typy matic

**ones** matice tvořená samými jedničkami  
**eye** jednotková matice  
**zeros** matice nul

### Základní operace s maticemi

Stručný seznam operátorů získáte pomocí **help ops**.

<b>C=A+B</b>	součet matic (sčítání stejnohlých)
<b>D=A-B</b>	rozdíl matic
<b>E=A*B</b>	součin matic (klasické násobení)
<b>F=A.*B</b>	násobení stejnohlých prvků matic
<b>G=A/B, G=A*inv(B)</b>	dělení matic zprava
<b>H=A./B</b>	podíl stejnohlých prvků matic <b>A</b> a <b>B</b>
<b>I=A\B, I=inv(A)*B</b>	dělení matic zleva
<b>J=A.\B</b>	podíl stejnohlých prvků matic <b>B</b> a <b>A</b>

## Základní maticové funkce - pomocí *help matfun*

Vybrané funkce:	<b>K=inv(A)</b>	inverzní matice k <b>A</b>
	<b>L=A'</b>	transpozice matice
	<b>det(A)</b>	determinant čtvercové matice

## 6. Práce s mnohočleny

Zobrazí se pomocí *help poly*.

<b>conv</b>	násobení mnohočlenů
<b>deconv</b>	dělení mnohočlenů
<b>poly</b>	mnohočleny pomocí kořenů
<b>residue</b>	výpočet residuí
<b>roots</b>	výpočet kořenů

Příklady:	<code>poly=[2 1 3 10]</code>	pro polynom $y=2x^3 + x^2 + 3x + 10$
	<code>koreny1=roots(poly)</code>	výpočet kořeny z daného mnohočlenu
	<code>koreny2=roots[2 1 3 10]</code>	dtto
	<code>poly1=poly(koreny1)</code>	výpočet koeficientů mnohočlenu z kořenů

## 7. Grafika 2D, 3D

<b>figure</b>	vytvoření grafického okna
<b>plot</b>	vytvoření 2D grafu
<b>plot3</b>	vytvoření spojitého 3D grafu
<b>mesh</b>	vytvoření 3D plošného grafu
<b>surf</b>	- dtto-
<b>bar, bar3</b>	2D, 3D sloupcový graf
<b>area</b>	plošný graf
<b>pie, pie3</b>	2D, 3D koláčový graf
<b>stairs</b>	schodišťový graf
<b>stem, stem3</b>	2D, 3D stopkový graf

Vyzkoušejte: **figure, Help, Demos** nebo pomocí **graf2d, graf2d2, hndlgraf, graf3d, hndlaxis**

```
t = 0:0.1:10;  
y = sin(t);  
plot(t,y)
```

```
y = sin(t) + sin(3*t)/3;  
plot(t,y)
```

```
y = sin(t) + sin(3*t)/3 + sin(5*t)/5 + sin(7*t)/7 + sin(9*t)/9;  
plot(t,y)
```

```
t = 0:0.02:3.14;  
y = zeros(10,length(t));  
x = zeros(size(t));  
for k=1:2:19  
    x = x + sin(k*t)/k;  
    y((k+1)/2,:) = x;  
end  
plot(y(1:2:9,:))  
title('název')
```

### Porovnejte:

a) `figure(10)`  
`t=0:0.1:10; y=sin(t); plot(t,y)`  
`figure(20)`  
`t=0:0.1:10; y=cos(t); plot(t,y)`

b) `figure(10)`  
`t=0:0.1:10; y=sin(t); plot(t,y);`  
`figure(20);`  
`t=0:0.1:10; y=cos(t); plot(t,y);`  
`hold on`  
`z=sin(t);`  
`plot(t,z)`

více křivek v jednom obrázku

c) `subplot(a,b,c)`

více obrázků v jednom grafickém okně  
**a** je počet obrázků vodorovně  
**b** je počet obrázků svisle  
**c** je určení pozice v matici **a\*b**

Více viz *help plot*, *help plot3*

## 8. Pro pokročilejší studenty

Kromě interaktivního způsobu práce lze k zápisu příkazů a povelů používat tzv. **m-soubory**. Jsou to textové soubory s logickým sledem příkazů, jsou uloženy např. na disku, **mají povinnou příponu \*.m**. Spouští se přímo v prostředí MATLAB napsáním jména souboru bez přípony \*.m.

Používáme-li vestavěný editor (vyvolá se příkazem *edit*), systém přípony \*.m připojí sám.

Používají se běžné programátorské povely. V případě syntaktických chyb jsou vypsána chybová hlášení s odkazem na příslušný řádek a s označením chyby. K rychlému nalezení příslušného řádku slouží **Ctrl-G**.

### **Příklad použití for, if, if-else, switch, case, while, break.**

Příklad 1:

```
% Definice matice 3x3
% zamena sloupce matice
A=[1 2 3; 4 5 6; 10 12 0.1]; % čtvercová matice
[a,b]=size(A); % určení počtu radku a sloupce
B=zeros(a,b); % prazdna matice stejnych rozmeru
for k=1: a
    for m=1: b
        B(k,m)=A(k,b-m+1); % cyklus pro zamenu sloupce
    end
end
A % vypis puvodni matice a
B % vypis nove matice B
```

Příklad 2:

```
if I == J
    A(I, J) = 2;
elseif abs(I-J) == 1
    A(I, J) = -1;
else
    A(I, J) = 0;
end
```

Příklad 3:

```
% prepínac SWITCH a CASE
for k=1: 4
    cislo = k;
    switch cislo
        case(1)
            disp('cislo ma hodnotu 1');
        case(2)
            disp('cislo ma hodnotu 2');
        otherwise
            disp('cislo ma hodnotu 3');
    end
end
% prikaz WHILE
Suma=0;
while Suma < 3
    Suma=Suma+1
end
% prikaz BREAK k zastaveni cyklu
for k=1 : 10
```



```
k
if k == 2
    break
end
end
disp('Preruseni algoritmu')
```

Více informací naleznete v [5].

### **Doporučené publikace:**

[1] <http://www.mathworks.com>

[2] <http://www.humusoft.cz>

[3] Dušek,F.: MATLAB a SIMULINK úvod do používání. Skripta, Univerzita Pardubice, 2002, ISBN 80-7194-475-0.

[4] Zaplatílek,K., Doňar,B.: MATLAB pro začátečníky. BEN-technická literatura, Praha 2003, 1.vydání, ISBN 80-7300-095-4.

[5] Zaplatílek,K., Doňar,B.: MATLAB - tvorba uživatelských aplikací. BEN-technická literatura, Praha 2004, 1.vydání, ISBN 80-7300-133-0.